Trapping rain water

Given an array arr[] of N non-negative integers representing the height of blocks. If width of each block is 1, compute how much water can be trapped between the blocks during the rainy season.

Example 1:Input:
N = 6
arr[] = {3,0,0,2,0,4}

Output:
10
Explanation:
Example 2:Input:
N = 4
arr[] = {7,4,0,9}
Output:
10
Explanation:
Water trapped by above
block of height 4 is 3 units and above
block of height 0 is 7 units. So, the
total unit of water trapped is 10 units.

Example 3:Input:
N = 3
arr[] = {6,9,9}
Output:
0
Explanation:
No water will be trapped.

PROGRAM:

```
package com.company;
import java.util.*;
class Main {
    public static int trap(int[] bars) {
        int n = bars.length;
            if (n <= 2) {
                return 0;
            }
            int water = 0;
            int[] left = new int[n-1];
            left[0] = Integer.MIN_VALUE;
            for (int i = 1; i < n - 1; i++) {
                left[i] = Integer.max(left[i - 1], bars[i - 1]);
            }
            int right = Integer.MIN_VALUE;
            for (int i = n - 2; i >= 1; i--)
            {
                right = Integer.max(right, bars[i + 1]);
                if (Integer.min(left[i], right) > bars[i]) {
                    water += Integer.min(left[i], right) - bars[i];
                }
            }
            return water;
```

```
        }
        public static void main(String[] args) {
        int[] heights = { 7, 0, 4, 2, 5, 0, 6, 4, 0, 5 };
            System.out.print("The maximum amount of water that can be
trapped is " + trap(heights));
        }
    }
```

OUTPUT :